# A Novel Classification via Clustering Method for Anomaly Based Network Intrusion Detection System

Mrutyunjaya Panda[1] and Manas Ranjan Patra[2]
[1] Department of ECE, Gandhi Institute of Engineering and Technology, Gunupur, Orissa-765022, India
Email: mrutyunjaya.2007@rediffmail.com
[2] Department of Computer Science, Berhampur University, Orissa-760007, India
Email: mrpatra12@gmail.com

***Abstract*- Intrusion detection in the internet is an active area of research. Intruders can be classified into two types, namely; external intruders who are unauthorized users of the computers they attack, and internal intruders, who have permission to access the system but with some restrictions. The aim of this paper is to present a methodology to recognize attacks during the normal activities in a system. A novel classification via sequential information bottleneck (sIB) clustering algorithm has been proposed to build an efficient anomaly based network intrusion detection model. We have compared our proposed method with other clustering algorithms like X-Means, Farthest First, Filtered clusters, DBSCAN, K-Means, and EM (Expectation-Maximization) clustering in order to find the suitability of our proposed algorithm. A subset of KDDCup 1999 intrusion detection benchmark dataset has been used for the experiment. Results show that the proposed method is efficient in terms of detection accuracy, low false positive rate in comparison to the other existing methods.**

***Index Terms*: Intrusion Detection, Meta Classifier, Unsupervised clustering, Sequential Information bottleneck clustering.**

## I INTRODUCTION

Identifying unauthorized use, misuse and attacks on information systems is defined as intrusion detection [1, 2]. The most popular way to detect intrusions has been done by analyzing audit data generated by operating systems and by networks. Since almost all activities are logged on a system, it is possible that a manual inspection of these logs would allow intrusions to be detected. It is important to analyze the audit data even an attack has occurred, for determining the extent of damage occurred, this analysis helps in attack trace back and also in recording the attack patterns for future prevention of such attacks. An intrusion detection system (IDS) can be used to analyze audit data for such insights. This makes IDS a valuable real-time detection and prevention tool as well as a forensic analysis tool.

Traditionally, intrusion detection techniques fall into two categories: signature detection and anomaly detection. Signature detection, or misuse detection, searches for well known patterns of attacks and intrusions by scanning for pre-classified signatures in network packets. On the other hand, anomaly detection can detect new intrusions while misuse detection may not. The idea behind anomaly detection is that if we can establish a normal activity profile for a system, in theory we can flag all system states varying from the established profile as intrusion attempts. However, if a set of intrusive activities is not identical to the set of known anomalous activities, the situation becomes more interesting as we can find new interesting possibilities. Anomalous activities that are not intrusive but flagged as intrusive are referred as false positives. Actual intrusive activities that go undetected are called false negatives. This is a serious problem and far more serious than false positives. Anomaly detection is computationally expensive because of the overhead of keeping track of and possibly updating several system profiles. Recently proposed system learning rules for anomaly detection (LERAD) discovers relationships among attributes in order to model application protocols [3, 4].

In fact, intrusion detection can be regarded as a classification problem, namely; identifying normal and other types of intrusive behavior. Hence, the key point here is to choose an effective classification approach to build accurate intrusion detection models. A number of machine learning algorithms have been applied to intrusion detection to learn intrusion rules or build normal usage patterns. In this paper, we investigate and evaluate the performance of classifiers using different cluster algorithms like Sequential Information Bottleneck (sIB), Farthest First Traversal (FFT), K-Means, DBSCAN and EM. The motivation behind using the classification using clustering algorithms is to improve the accuracy of the intrusion detection system in comparison to the individual clustering algorithms. The rest of the paper is organized as follows. A literature review is presented in Section 2 followed by a short theoretical background on the clustering algorithms used in this research in Section 3. Section 4 provides a brief idea about the KDDCup 1999 dataset used. The proposed methodology is described in Section 5 followed by the experimental setup in Section 6. Section 7 describes the results and




DOI: 01.ijns.01.02.04

discussion. Finally, the paper concludes with future research directions in Section 8.

## II. RELATED RESEARCH

In [5], the authors have designed an Intrusion response (IR) system cooperating with IDS using mobile agents distributed throughout the network, based on stigmergic properties. In [6], the authors introduced a self-organized ant colony based intrusion detection system (ANTIDS) to detect intrusions and compares its performance with linear genetic programming (LGP) [7], Support vector machines (SVM) [8] and Decision Trees (DT) [9]. Other works have made use of Multiple Adaptive Regression Splines (MARS) [10]. In [11], the authors have compared various data mining algorithms for detecting network intrusions. The authors have used Naïve Bayes algorithm in building a network intrusion detection model [12]. In [13], the authors proposed Bayesian Belief network (BBN) with genetic and simulated annealing local search in order to build an efficient network intrusion detection model. The authors have compared various ensemble algorithms in detecting the intrusion detection in [14]. Modeling intrusion detection system using hybrid intelligent systems is proposed in [15]. In this, DT and SVM are combined as a hierarchical hybrid intelligent system model (DT_SVM) and an ensemble approach combining the base classifiers. In [16], the authors propose support vector learning approach to classify network requests. The authors in [17] used K-Means and DBSCAN to demonstrate how cluster analysis can be used to effectively identify groups of traffic that are similar using only transport layer statistics. The authors propose hierarchical Gaussian Mixture Model (HGMM) a novel type of Gaussian Mixture which detects network based attacks as anomalies using statistical processing classification in [18]. In [19], the authors use automated feature weighting for network anomaly detection. They conclude that their proposed method not only increases the detection rate but also reduces false alarm rate as well.

We believe that an unsupervised clustering approach offers some advantages over supervised learning approaches. One of the main benefits is that new applications can be identified by examining the connections that are grouped to form a new cluster. The supervised approach can not discover new applications and can only classify network traffic for which it has labeled training data. This prompted us to use unsupervised clustering approach in building an efficient classifier for detecting anomaly based network intrusions.

## I. CLUSTERING ALGORITHMS

In our experiments, some clustering algorithms are selected for performance comparison. Some of them are discussed as follows.

### A. K-Means:

We first describe the K-means algorithm for producing a clustering of the points in the input into K clusters. It partitions the data-points into K subsets such that all points in a given subset "belong" to some centre. The algorithm keeps track of the centroids of the subsets and proceeds in iterations. Before the first iteration the centroids are initialized to random values. The algorithm terminates when the centroid locations stay fixed during iteration. During each iteration the following is performed:

- For each point x, find the centroid which is closest to x. Associate x with this centroid.
- Re-estimate centroid locations by taking, for each centroid, the canter of mass of points associated with it.

The K-Means algorithm is known to converge to a local minimum of the distortion measure (that is, average squared distance from points to their class centroids). It is also known to be very simple and can be easily implemented in solving many practical problems. It can work very well for compact and hyper spherical clusters. The time complexity of the K-means is O (NKd). Since, K and d are usually much less than N, K-means can be used to cluster large data sets.

### B. Expectation Maximization (EM)

We follow the procedure as in [20]. The steps for our implementation of EM are as follows. We initiate the process by an initial estimate of mean and standard deviation. The EM algorithm then searches for a maximum likelihood hypothesis through the following iterative scheme.

- Initialization Step: Initialize the hypothesis

$$\theta^0 = \left( \mu_1^0, \mu_2^0, ...., \mu_k^0 \right), \ \theta_k^0 = \mu_k^0 \quad (1)$$

Where k is the current number of Gaussians, $\sigma$ is the standard deviation, $\theta^0$ is the estimate at 0th iteration, $\mu$ is the mean.

- Expectation Step: Estimate the expected values of the hidden variables $z_{ij}$ (mean and standard deviation) using the current hypothesis $\theta^t = \left( \mu_1^t, \mu_2^t, ...., \mu_k^t \right)$

$$E(z_{ik}) = \frac{\exp\left[ \frac{(x_i - \mu_k)^2}{2\sigma^2} \right]}{\sum_{j=1}^{k} \exp\left[ \frac{(x_i - \mu_j)^2}{2\sigma^2} \right]} \quad (2)$$

where t is the number of iteration, $E\left( z_{ij} \right)$ is the expected value for the hidden variables (namely




DOI: 01.ijns.01.02.04

mean and standard deviation), k is the dimension, $\sigma$ is the standard deviation.

- Maximization Step: provides a new estimate of the parameters.

$$\mu_k^{t+1} = \frac{\sum_{i=1}^{n} E(z_{ik})\, x_i}{\sum_{i=1}^{n} E(z_{ik})} \quad (3)$$

- Convergence Step: if $\left\|\theta^{t+1} - \theta^t\right\| < e$, stop (finish iteration); otherwise, go to step 2.

The hidden variables are the parameters of the model. In this case, we use mixture of Gaussians; hence the hidden variables are the mean and standard deviation for each Gaussian distribution. We start with an initial estimate of those parameters and iteratively run the algorithm to find the maximum likelihood (ML) for our estimates.

The reason we are using EM is to fit the data better, so that clusters are compact and far from other clusters, since we initially estimate the parameters and iterate to find the ML for those parameters

C. DBSCAN

Density-based methods are based on a simple assumption: clusters are dense regions in the data space that are separated by regions of lower density. Their general idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold. In other words, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. These methods are good at altering out outliers and discovering clusters of arbitrary shapes.

Density based spatial clustering of applications with noise (DBSCAN) is an example of density based methods [21].

The algorithm DBSCAN (Density Based Spatial Clustering of Applications with Noise) targeting low-dimensional spatial data is the major representative in this category. Two input parameters $\varepsilon$ and MinPts are used to define:

1)An $\varepsilon$ -neighborhood $N_\varepsilon(x) = y\ \Sigma X \mid d(x,y)\ \ \varepsilon$ ) of the point x,

2) A core object (a point with a neighborhood consisting of more than MinPts points)

3) A concept of a point y density-reachable from a core object x (a finite sequence of core objects between x and y exists such that each next belongs to an $\varepsilon$ -neighborhood of its predecessor)

4) A density-connectivity of two points x, y (they should be density-reachable from a common core object).

More details about the DBSCAN can be found in [22].

D. Farthest First

The farthest first traversal k-center algorithm (FFT) is a fast, greedy algorithm that minimizes the maximum cluster radius [23]. This is also treated as an efficient algorithm which always returns the right answer. The pseudo code for the farthest first traversal algorithm is as follows:

- Pick any $z \in S$ and set $T = \{z\}$
- While

$$|T| < k:$$
$$z = \arg\max_{x \in S} \rho(x,T)$$
$$T = T \cup \{z\}$$

Here, $\rho(x,T)$ is the distance from point $(x)$ to the closest point in set T. This builds a solution T one point at a time. It starts with any point, and then iteratively adds in the point farthest from the ones chosen so far. The farthest point $(x)$ from a set $(S)$ is obtained from $\rho(x,T)$. Farthest first traversal takes time $O(k|S|)$, which is fairly efficient and is always close to optimal solution, in the sense that if T is the solution returned by the farthest first traversal, and $T^*$ is the optimal solution, then $\cos t(T) \le 2\cos t(T^*)$ .

E. Filtered Clusterer

The Filtered Clusterer is Meta-Clusterer which offers the possibility to apply filters directly before the Clusterer is learnt. The structure of the filter is based exclusively on the training data and test instances are processed by the filter without changing their structure [24].

F. X-Means

X-means is a new algorithm that quickly estimates the number of clusters K. It goes into action after each run of K-means, making local decisions about which subset of the current centroids should split themselves in order to better fit the data. The splitting decision is done by computing the Bayesian Information Criterion (BIC). X-means have been experimented against a more traditional method that estimates the number of clusters by guessing K. X-means consistently produced better clustering on both synthetic and real-life data, with respect to BIC. It also runs much faster, which prompted us to select this algorithm for our intrusion detection model, which has not been considered so far for intrusion detection. A detailed description on X-Means operations can be seen in [25].

## IV INTRUSION DETECTION DATASET

The KDD Cup 1999 Intrusion detection contest data is used in our experiments. This data was prepared by DARPA Intrusion detection evaluation program by




DOI: 01.ijns.01.02.04

MIT Lincoln Laboratory. Lincoln labs acquired nine weeks of raw TCP dump data. The raw data was processed into connection records, which contains about 5 million connection records. The data set contains 24 attack types. These attacks fall into four main categories such as Denial of Service (DoS), Probing, User to Root (U2R) and Remote to Local (R2L) attacks.

Besides the four different types of attacks mentioned above, normal class needs to be detected. The data set for our experiments contained 1000 connection records, which is a subset of 10% KDD Cup'99 intrusion detection benchmark dataset. These were randomly generated from the MIT data set. Random generation of data include the number of data from each class proportional to its size, except that the smallest class is completely included. All the intrusion detection models are trained and tested with the same data set. Although, the data set contains five different classes, we perform 2-class classification (either normal or attack) as our proposed method could not handle multiple class labels for building an efficient intrusion detection model.

## V PROPOSED METHODOLOGY

The proposed methodology for classification Via Clustering using Sequential Information Bottleneck (sIB) principle is shown in Fig. 1 below.

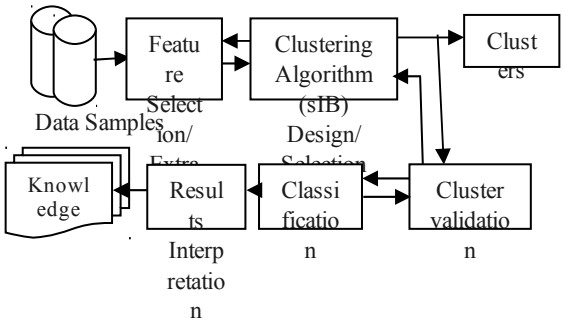


Figure1.Proposed methodology of Classification Via sIB clustering

### A. Meta Classifier

Meta Classifier which comes under Decision committee learning has demonstrated spectacular success in reducing classification error from learned classifiers. These techniques develop a classifier in the form of a committee of subsidiary classifiers. The committee members are applied to a classification task and their individual outputs combined to create a single classification from the committee as a whole. These are recent methods for improving the predictive power of classifier learning systems.

In our proposed methodology, we design a simple meta-classifier that uses a clusterer for classification. For cluster algorithms that use a fixed number of Clusterer, like Simple K-Means, the user has to make sure that the number of clusters to generate is the same as the number of class labels in the dataset in order to obtain a useful model. However, in case of sIB clustering algorithm, we need not specify the number of clusters to generate a model, as it fixes the number of clusters automatically.

### B. Proposed Sequential Information Bottleneck Clustering(sIB)

The steps for the Sequential Information Bottleneck (sIB) Clustering are as follows.

- Finding the global minimum for given number of clusters (W)
- Initialized with a (possibly random) partition of W clusters
- sIB draws a sample $x$ at random, treats it as a singleton cluster and merges it to a new cluster $c_{new}$ such that,

$$c_{new} = \arg\min_{c \in C} JS(x, c) \qquad (4)$$

Where $JS(.,.)$ is the Jensen-Shannon distance.

- At each step the objective function improves or remains un-changed.

More details about this algorithm can be found in [26].

## VI EXPERIMENTAL SETUP

All our experiments were performed using a Pentium 4, 2.8 GHz CPU with 512 MB RAM. Full data set is used for the purpose of training in order to build an intrusion detection model. Then, 10-fold cross validation method is used in order to test the efficacy of the model built during the training phase.

### A. Cross Validation Methods

Cross-validation (CV) tests exist in a number of variants but the general idea is to divide the training data into a number of partitions or folds. The classifier is evaluated by its classification accuracy over one partition after having learned from the other. This procedure is then repeated until all partitions have been used for evaluation. Some of the most common types are 10-fold, n-fold and bootstrap CV [27]. The difference between these three types of CV lies in the way data is partitioned. Leave-one-out is similar to n-fold CV, where n stands for the number of instances in the data set. Leave–one-out or n-fold CV is performed by leaving one instance out for testing and training on the other instances. This procedure is then performed until all instances have been left out once. It has been argued that the design of 10-fold cross-validation introduces a source of correlation since one is used for training in one trial and then the same is used for testing in another [28].

## VII RESULTS AND DISCUSSION

Here, we present the behavior of different unsupervised approaches in building an efficient anomaly based network intrusion detection model. In Table 1, we




DOI: 01.ijns.01.02.04

present a comparative study of our proposed sIB clustering method with respect to the other existing approaches.

Table 1. Comparative Study of Existing Clustering and Classification Algorithms

| Algorithm | Average Detection Rate (%) | False Positive Rate (%) |
|---|---|---|
| K-Means [29] | 46.986 | 0.875 |
| KM-VQ [19] | 48.4 | 10 |
| FES-KM-VQ[19] | 60.1 | 10 |
| X-Means (ours) | 78 | 36 |
| KDDCup Winner [29] | 48.57 | 0.225 |
| GMIX [30] | 53.725 | 29.715 |
| SOM [30] | 47 | 25.685 |
| Nearest Cluster [31] | 47.875 | 0.2 |
| Incremental Clustering [32] | 44.57 to 84.78 | 15.99 to 76.83 |
| Cluster [32] | 66 | 2 |
| K-NN [32] | 23 | 6 |
| FR-1 [33] | 74.82 | 73.07 |
| FR-2 [33] | 79.68 | 85.66 |
| Pure SVM [34] | 57.6 | 35.5 |
| SVM + Rocchio bundling [34] | 51.6 | 44.2 |
| SVM +DGSOT [34] | 69.8 | 37.8 |
| sIB (Proposed) | 85.5 | 34 |
| ANTIDS-*a* [6] | 69.9 | ---- |

From, the above comparison, it is clear that our proposed method is efficient in detecting anomalous activities with high detection rate and relatively low false positive rate.

It is also quite evident from Table 1 and Table 2 that the Meta classifier using sIB clustering for classification enhances the intrusion detection rate while maintaining the same false positive rate in comparison to unsupervised sIB clustering method.

It can be seen from Table 2 that our proposed method is able to detect intrusions with highest accuracy in comparison to K-Means and Filtered Clusterer with K-Means classification via clustering methods. It can also be observed that the proposed methodology with X-Means, FFT, DBSCAN and EM clustering provides high average detection rate in comparison to our sIB clustering, but they are not able to classify all the instances successfully. At the same time, their false positive rates except in case of X-Means are more than sIB clustering. These comparisons show that our proposed classification via sIB clustering is more suitable in building an efficient anomaly based network intrusion detection model.

Table 2.Comparison of our proposed Classification via Clustering Methods

| Classification via Clustering Algorithms | Average Detection Rate (%) | Average False positive Rate (%) | Unclassified instances (%) |
|---|---|---|---|
| K-Means | 80.1 | 35 | NIL |
| X-Means | 88.3 | 26.8 | 12.6 |
| Filtered Clusterer +K-Means | 80.1 | 19.8 | NIL |
| Farthest First (FFT) | 86.6 | 89.6 | 0.8 |
| DBSCAN | 86.7 | 100 | 18.66 |
| EM | 89.1 | 52.3 | 2.2 |
| sIB (Proposed) | 86.3 | 34 | NIL |

## VIII. CONCLUSION AND FUTURE SCOPE

In this paper, we have illustrated the use of classification via clustering methodology using sIB clustering algorithm for modeling intrusion detection systems. The proposed approach provides better detection accuracy with comparatively low false positive rate in comparison to other existing unsupervised clustering algorithms. This makes the approach suitable for building an efficient anomaly based network intrusion detection model. As evident from the results none of the algorithms provide the best detection with zero false positive rates. Therefore, in our future research we shall investigate other data mining techniques with a view to enhance the detection accuracy as close as possible to 100% while maintaining a low false positive rate.

## REFERENCES

[1] D. Denning. An Intrusion detection model. IEEE Transaction on software Engineering.SE-13(2), pp.222-232, 1998.

[2] S. Kumar, E. H. Spafford. An application of pattern matching in Intrusion Detection. Technical report,CSD-TR-94-013. Purdue University, 1994.

[3] M.Mahoney, P.K.Chan. An analysis of the 199 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. In 6th Intl. symposium on RAID, 2003. pp. 220-237.

[4] P. K. Chan, M. mahoney and M. Arshad. Learning rules and clusters for anomaly detection in network traffic", in managing cyber threats,issues,approaches and challenges, V.Kumar,J.Srivastava, and A.Lazarevic (Edns.),Kluwer.2004.

[5] N. Foukia, S. Hassas, S. Fenet and J. Hulaas, "An Intrusion response scheme:Tracking the source using the sigmery paradigm", in Proc. Of security of mobile multi agent system workshop (SEMAS-2002).Italy, July16, 2002.

[6] V. Ramos and Ajith Abraham, "ANTIDS-Self organised Ant based clustering model for intrusion detection system. WSTST, 2005, pp.103-112. http://www.softcomputing.net/WSTST-ra.pdf




DOI: 01.ijns.01.02.04

[7] M. Brameier and W. Banzhaf, "A comparison of linear genetic programming and neural network in medical data mining", in IEEE Transaction on Evolutionary computation.5 (1), pp.17-26, 2001.

[8] V.N. Vapnik, The Nature of Statistical Learning Theory. *Springer*, 1995.

[9] J. Denebourg ,et al. "The dynamic of collective sorting Robot like ants and Ant like Robots", in 1st conf. on simulation of Adaptive behaviour ;from animals to animats,cambridg,MA,MIT Press,pp.356-365,1991.

[10] S. Mukkamala, A.H. Sung, A. Abraham, V. Ramos, "Intrusion Detection Systems using Adaptive Regression Splines", in ICEIS-04, *6th Int. Conf. on Enterprise Information Systems*, to appear at Kluwer Academic Press, 2005, Porto, 14-17 April 2004.

[11] M. Panda and M. R. Patra, "A comparative study of data mining algorithms for network intrusion detection", proc. of ICETET, India, 2008.pp.504-507. IEEE Xplore.

[12] M. Panda and M. R. Patra, "Network intrusion detection using Naïve Bayes", International journal of computer science and network security, vol.7, No.12, 2007, pp.258-263.

[13] M. Panda and M. R. Patra. "Bayesian belief Network using genetic local search for network intrusion", International journal of secure digital information age.Vol.1, issue.1, June 2009. In Press.

[14] M. Panda and M. R. Patra, "Network intrusion detection using boosting support vector classifiers", In 2009 IEEE Intl.Advance computing Conference, ptaila, Punjab, pp.926-931. IEEE Press.USA.

[15] S. Peddabachigari, A. Abraham, C. Grosan and J. Thomas, "Modelling intrusion detection system using hybrid intelligent systems", Journal of Network and computer applications.Elsevier, 30(1), pp.114-132, 2007.

[16] John Mill and A. Inoue, "Support vector classifier and network intrusion detection", In proc. of 2004 IEEE international conference on fuzzy system, WA, USA, 2004, Vol.1, pp.407-410. ISBN: 1098-7584.

[17] J. Erman, M. Arlitt and A. Mahanti, "Traffic classification using clustering algorithms", in SIGCOMM-06 workshops, sept.11-15, 2006, Pisa, Italy.pp.281-286.ACM Press.

[18] M. Bahrololum and M. Khaleghi, "Anomaly intrusion detection system using hierarchical gausian mixture model", International journal of computer science and network security, vol.8, no.8, pp.264-271, August 2008.

[19] Dat Tran, W. Ma and Dharmendra Sharma, "Automatedfeature weighting for network anomaly detection", in International journal of computer science and network security, Vol.8, no.2, pp.173-178, Feb.2008.

[20] The Expectation Maximization Algorithm. http://www.cs.unr.edu/~bebis/mathmethods/EM/lecture.pdf. Sept. 25, 2004.

[21] M. Ester, H.-P Kriegel, J. Sander and X. Xu, „A density-based algorithm for discovering clusters in large spatial database with noise", in Proc. of KDD-96,pp.226-231,1996.

[22] Pavel Berkhin, "Survey of clustering data mining techniques", Technical report. Accrue software, San Jose,CA,USA,pp.1-56.2002. http://www.ee.ucr.edu/~barth/EE242/clustering_survey.pdf

[23] A.Willium, "Clustering algorithm for categorical data".2006. http://www.cse.yorku.ca/~billa/MULIC/Bill_Andreopoulous_PhD_thesis_web.pdf

[24] A. William, "Clustering Algorithms for categorical data", September 2006. http://www.cse.yorku.ca/~billa/MULIC/Bill_Andreopoulos_PhD_thesis_web.pdf

[25] D. Pellag and A. Moore, "X-Means: Extending K-Means with efficient estimation of the number of clusters". Int. Conf. on Machine Learning (ICML), pp.727-734, San Francisco, 2000.

[26] N. Slonim, N. Friedman and N. Tishby, "Unsupervised document classification using sIB maximization: SIGIR, Tempere, Finland, pp.129-136, ACM Press. ISBN: 1-58113-561-0102/0008.

[27] N. Lavesson and P. Davidson, "Multi dimensional measures function for classifier performance", in 2nd IEEE Intl. Conf. on Intelligent systems, June 2004,pp.508-513.

[28] M. A. Maloof, "On machine learning ROC analysis and statistical tests of significance", in 16th Intl.Conf. on pattern recognition, IEEE,Vol.2,2002,pp.204-207.

[29] I. Levin,"KDD-99 classifier learning context", in LLSoft's results overview, SIGKDD explorations, ACMSIGKDD, Vol.1, No.2, pp.67-75, 2000.

[30] V. Venkatchalam and S. Selvan, "Performance comparison of intrusion detection system classifier using various feature reduction techniques", International journal of simulation, Vol.9, No.1, 2008.

[31] M. K. Sabhani and G. Sarpen, "Application of machine learning algorithm on KDD intrusion detection dataset with misuse detection context", in Proc. of Intl.conf. on machine learning models,technologies and applications (MLMTA 2005), Los Vegas,NV,2003,pp.209-215.

[32] T. Hassan, M. Hashaem and A. Fahmy," Unsupervised anomaly detection using an incremental clustering algorithm", International journal of intelligent computing and information sciences, Vol.5, No.1, pp.253-268, 2005.




DOI: 01.ijns.01.02.04